uCertify

Course Outline

Data Structures and Abstractions with Java

20 May 2024

1. Course Objective
2. Pre-Assessment
3. Exercises, Quizzes, Flashcards & Glossary

   Number of Questions
4. Expert Instructor-Led Training
5. ADA Compliant & JAWS Compatible Platform
6. State of the Art Educator Tools
7. Award Winning Learning Platform (LMS)
8. Chapter & Lessons

   Syllabus

   Chapter 1: Introduction

   Chapter 2: Prelude: Designing Classes

   Chapter 3: Bags

   Chapter 4: Bag Implementations That Use Arrays

   Chapter 5: A Bag Implementation That Links Data

   Chapter 6: The Efficiency of Algorithms

   Chapter 7: Stacks

   Chapter 8: Stack Implementations

   Chapter 9: Queues, Deques, and Priority Queues

   Chapter 10: Queue, Deque, and Priority Queue Implementations

   Chapter 11: Recursion

   Chapter 12: Lists

   Chapter 13: A List Implementation That Uses an Array

   Chapter 14: A List Implementation That Links Data

   Chapter 15: Iterators for the ADT List

   Chapter 16: Problem Solving with Recursion

   Chapter 17: An Introduction to Sorting

   Chapter 18: Faster Sorting Methods

1. Course Objective

The Data Structures and Abstractions with Java course covers a wide range of topics related to data structures and algorithms, including arrays, linked lists, stacks, queues, trees, graphs, recursion, sorting, sets, and maps. It provides introduction to these concepts, starting with the basics and gradually building up to more advanced topics. The course covers the implementation of data structures using the Java programming language, with a focus on design decisions and safe and secure programming practices. The course also covers the topics, such as Java basics, file input and output and glossary, making the course a comprehensive resource for learners seeking to enhance their knowledge in data structures and Java programming.

2. Pre-Assessment

Pre-Assessment lets you identify the areas for improvement before you start your prep. It determines what students know about a topic before it is taught and identifies areas for improvement with question assessment before beginning the course.

3. Exercises

There is no limit to the number of times learners can attempt these. Exercises come with detailed remediation, which ensures that learners are confident on the topic before proceeding.

4. Quiz

Quizzes test your knowledge on the topics of the exam when you go through the course material. There is no limit to the number of times you can attempt it.

5. flashcards

Flashcards are effective memory-aiding tools that help you learn complex topics easily. The flashcard will help you in memorizing definitions, terminologies, key concepts, and more. There is no limit to the number of times learners can attempt these. Flashcards help master the key concepts.

6. Glossary of terms

uCertify provides detailed explanations of concepts relevant to the course through Glossary. It contains a list of frequently used terminologies along with its detailed explanation. Glossary defines the key terms.

7. Expert Instructor-Led Training

uCertify uses the content from the finest publishers and only the IT industry's finest instructors. They have a minimum of 15 years real-world experience and are subject matter experts in their fields. Unlike a live class, you can study at your own pace. This creates a personal learning experience and gives you all the benefit of hands-on training with the flexibility of doing it around your schedule 24/7.

8. ADA Compliant & JAWS Compatible Platform

uCertify course and labs are ADA (Americans with Disability Act) compliant. It is now more accessible to students with features such as:

- Change the font, size, and color of the content of the course
- Text-to-speech, reads the text into spoken words
- Interactive videos, how-tos videos come with transcripts and voice-over
- Interactive transcripts, each word is clickable. Students can clip a specific part of the video by clicking on a word or a portion of the text.

JAWS (Job Access with Speech) is a computer screen reader program for Microsoft Windows that reads the screen either with a text-to-speech output or by a Refreshable Braille display. Student can easily navigate uCertify course using JAWS shortcut keys.

9. State of the Art Educator Tools

uCertify knows the importance of instructors and provide tools to help them do their job effectively. Instructors are able to clone and customize course. Do ability grouping. Create sections. Design grade scale and grade formula. Create and schedule assessments. Educators can also move a student from self-paced to mentor-guided to instructor-led mode in three clicks.

10. Award Winning Learning Platform (LMS)

uCertify has developed an award winning, highly interactive yet simple to use platform. The SIIA CODiE Awards is the only peer-reviewed program to showcase business and education technology's finest products and services. Since 1986, thousands of products, services and solutions have been

recognized for achieving excellence. uCertify has won CODiE awards consecutively for last 7 years:

- **2014**
    1. Best Postsecondary Learning Solution

- **2015**
    1. Best Education Solution
    2. Best Virtual Learning Solution
    3. Best Student Assessment Solution
    4. Best Postsecondary Learning Solution
    5. Best Career and Workforce Readiness Solution
    6. Best Instructional Solution in Other Curriculum Areas
    7. Best Corporate Learning/Workforce Development Solution

- **2016**
    1. Best Virtual Learning Solution
    2. Best Education Cloud-based Solution
    3. Best College and Career Readiness Solution
    4. Best Corporate / Workforce Learning Solution
    5. Best Postsecondary Learning Content Solution
    6. Best Postsecondary LMS or Learning Platform
    7. Best Learning Relationship Management Solution

- **2017**
    1. Best Overall Education Solution
    2. Best Student Assessment Solution
    3. Best Corporate/Workforce Learning Solution
    4. Best Higher Education LMS or Learning Platform

- **2018**
    1. Best Higher Education LMS or Learning Platform

2. Best Instructional Solution in Other Curriculum Areas
3. Best Learning Relationship Management Solution

- **2019**
  1. Best Virtual Learning Solution
  2. Best Content Authoring Development or Curation Solution
  3. Best Higher Education Learning Management Solution (LMS)

- **2020**
  1. Best College and Career Readiness Solution
  2. Best Cross-Curricular Solution
  3. Best Virtual Learning Solution

11. Chapter & Lessons

uCertify brings these textbooks to life. It is full of interactive activities that keeps the learner engaged. uCertify brings all available learning resources for a topic in one place so that the learner can efficiently learn without going to multiple places. Challenge questions are also embedded in the chapters so learners can attempt those while they are learning about that particular topic. This helps them grasp the concepts better because they can go over it again right away which improves learning.

Learners can do Flashcards, Exercises, Quizzes and Labs related to each chapter. At the end of every lesson, uCertify courses guide the learners on the path they should follow.

Syllabus

Chapter 1: Introduction

- Organizing Data

Chapter 2: Prelude: Designing Classes

- Encapsulation
- Specifying Methods
- Java Interfaces
- Choosing Classes
- Reusing Classes
- Exercises
- Projects

Chapter 3: Bags

- The Bag
- Specifying a Bag
- Using the ADT Bag
- Using an ADT Is Like Using a Vending Machine
- The ADT Set
- Java Class Library: The Interface Set
- Java Interlude 1: Generics
- Lesson Summary
- Programming Tip

- Exercises
- Projects

Chapter 4: Bag Implementations That Use Arrays

- Using a Fixed-Size Array to Implement the ADT Bag
- Using Array Resizing to Implement the ADT Bag
- The Pros and Cons of Using an Array to Implement the ADT Bag
- Java Interlude 2 Exceptions
- Lesson Summary
- Programming Tips
- Exercises
- Projects

Chapter 5: A Bag Implementation That Links Data

- Linked Data
- A Linked Implementation of the ADT Bag
- Removing an Item from a Linked Chain
- A Class Node That Has Set and Get Methods
- The Pros and Cons of Using a Chain to Implement the ADT Bag

- Lesson Summary
- Programming Tip
- Exercises
- Projects

Chapter 6: The Efficiency of Algorithms

- Motivation
- Measuring an Algorithm’s Efficiency
- Big Oh Notation
- Picturing Efficiency
- The Efficiency of Implementations of the ADT Bag
- Lesson Summary
- Exercises
- Projects

Chapter 7: Stacks

- Specifications of the ADT Stack
- Using a Stack to Process Algebraic Expressions

- The Program Stack
- Java Class Library: The Class Stack
- Lesson Summary
- Programming Tip
- Exercises
- Projects

Chapter 8: Stack Implementations

- A Linked Implementation
- An Array-Based Implementation
- A Vector-Based Implementation
- Java Interlude 3: More About Exceptions
- Lesson Summary
- Exercises
- Projects

Chapter 9: Queues, Deques, and Priority Queues

- The ADT Queue
- The ADT Deque

- The ADT Priority Queue
- Lesson Summary
- Programming Tip
- Exercises
- Projects

Chapter 10: Queue, Deque, and Priority Queue Implementations

- A Linked Implementation of a Queue
- An Array-Based Implementation of a Queue
- Circular Linked Implementations of a Queue
- Java Class Library: The Class AbstractQueue
- A Doubly Linked Implementation of a Deque
- Possible Implementations of a Priority Queue
- Lesson Summary
- Programming Tip
- Exercises
- Projects

Chapter 11: Recursion

- What Is Recursion?
- Tracing a Recursive Method
- Recursive Methods That Return a Value
- Recursively Processing an Array
- Recursively Processing a Linked Chain
- The Time Efficiency of Recursive Methods
- Tail Recursion
- Using a Stack Instead of Recursion
- Lesson Summary
- Programming Tips
- Exercises
- Projects

Chapter 12: Lists

- Specifications for the ADT List
- Using the ADT List
- Java Class Library: The Interface List
- Java Class Library: The Class ArrayList

- Lesson Summary
- Exercises
- Projects

Chapter 13: A List Implementation That Uses an Array

- Using an Array to Implement the ADT List
- The Efficiency of Using an Array to Implement the ADT List
- Lesson Summary
- Exercises
- Projects

Chapter 14: A List Implementation That Links Data

- Operations on a Chain of Linked Nodes
- Beginning the Implementation
- Continuing the Implementation
- A Refined Implementation
- The Efficiency of Using a Chain to Implement the ADT List
- Java Class Library: The Class LinkedList

- Java Interlude 4: Iterators
- Lesson Summary
- Exercises
- Projects

Chapter 15: Iterators for the ADT List

- Ways to Implement an Iterator
- A Separate Class Iterator
- An Inner Class Iterator
- Why Are Iterator Methods in Their Own Class?
- An Array-Based Implementation of the Interface ListIterator
- Lesson Summary
- Programming Tip
- Exercises
- Projects

Chapter 16: Problem Solving with Recursion

- A Simple Solution to a Difficult Problem
- A Poor Solution to a Simple Problem

- Languages and Grammars
- Indirect Recursion
- Backtracking
- Java Interlude 5: More About Generics
- Lesson Summary
- Exercises
- Projects

Chapter 17: An Introduction to Sorting

- Organizing Java Methods That Sort an Array
- Selection Sort
- Insertion Sort
- Shell Sort
- Comparing the Algorithms
- Lesson Summary
- Programming Tip
- Exercises
- Projects

Chapter 18: Faster Sorting Methods

- Merge Sort
- Quick Sort
- Radix Sort
- Comparing the Algorithms
- Java Interlude 6: Mutable and Immutable Objects
- Lesson Summary
- Exercises
- Projects

Chapter 19: Sorted Lists

- Specifications for the ADT Sorted List
- A Linked Implementation
- An Implementation That Uses the ADT List
- Java Interlude 7: Inheritance and Polymorphism
- Lesson Summary
- Exercises

- Projects

Chapter 20: Inheritance and Lists

- Using Inheritance to Implement a Sorted List
- Designing a Base Class
- An Efficient Implementation of a Sorted List
- Lesson Summary
- Programming Tips
- Exercises
- Projects

Chapter 21: Searching

- The Problem
- Searching an Unsorted Array
- Searching a Sorted Array
- Searching an Unsorted Chain
- Searching a Sorted Chain
- Choosing a Search Method
- Java Interlude 8: Generics Once Again

- Lesson Summary
- Programming Tip
- Exercises
- Projects

Chapter 22: Dictionaries

- Specifications for the ADT Dictionary
- Using the ADT Dictionary
- Java Class Library: The Interface Map
- Lesson Summary
- Programming Tips
- Exercises
- Projects

Chapter 23: Dictionary Implementations

- Array-Based Implementations
- Linked Implementations
- Lesson Summary

- Programming Tips
- Exercises
- Projects

Chapter 24: Introducing Hashing

- What Is Hashing?
- Hash Functions
- Resolving Collisions
- Lesson Summary
- Exercises
- Projects

Chapter 25: Hashing as a Dictionary Implementation

- The Efficiency of Hashing
- Rehashing
- Comparing Schemes for Collision Resolution
- A Dictionary Implementation That Uses Hashing
- Java Class Library: The Class HashMap
- Java Class Library: The Class HashSet

- Lesson Summary
- Exercises
- Projects

Chapter 26: Trees

- Tree Concepts
- Traversals of a Tree
- Java Interfaces for Trees
- Examples of Binary Trees
- Examples of General Trees
- Lesson Summary
- Exercises
- Projects

Chapter 27: Tree Implementations

- The Nodes in a Binary Tree
- An Implementation of the ADT Binary Tree
- An Implementation of an Expression Tree

- General Trees
- Using a Binary Tree to Represent a General Tree
- Java Interlude 9: Cloning
- Lesson Summary
- Programming Tips
- Exercises
- Projects

Chapter 28: A Binary Search Tree Implementation

- Getting Started
- Searching and Retrieving
- Traversing
- Adding an Entry
- Removing an Entry
- The Efficiency of Operations
- An Implementation of the ADT Dictionary
- Lesson Summary
- Exercises

- Projects

Chapter 29: A Heap Implementation

- Reprise: The ADT Heap
- Using an Array to Represent a Heap
- Adding an Entry
- Removing the Root
- Creating a Heap
- Heap Sort
- Lesson Summary
- Exercises
- Projects

Chapter 30: Balanced Search Trees

- AVL Trees
- 2-3 Trees
- 2-4 Trees
- Red-Black Trees
- B-Trees

- Lesson Summary
- Exercises
- Projects

Chapter 31: Graphs

- Some Examples and Terminology
- Traversals
- Topological Order
- Paths
- Java Interfaces for the ADT Graph
- Lesson Summary
- Exercises
- Projects

Chapter 32: Graph Implementations

- An Overview of Two Implementations
- Vertices and Edges
- An Implementation of the ADT Graph

- Lesson Summary
- Exercises
- Projects

Chapter 33: Appendix A: Documentation and Programming Style

- Naming Variables and Classes
- Indenting
- Comments

Chapter 34: Appendix B: Java Classes

- Objects and Classes
- Using the Methods in a Java Class
- Defining a Java Class
- Enumeration as a Class
- Packages

Chapter 35: Appendix C: Creating Classes from Other Classes

- Composition
- Inheritance

- Type Compatibility and Superclasses

Chapter 36: Supplement 1: Java Basics

- Introduction
- Elements of Java
- Simple Input and Output Using the Keyboard and Screen
- The if-else Statement
- The switch Statement
- Enumerations
- Scope
- Loops
- The Class String
- The Class StringBuilder
- Using Scanner to Extract Pieces of a String
- Arrays
- Wrapper Classes

Chapter 37: Supplement 2: File Input and Output

- Preliminaries

- Text Files
- Binary Files

12. Practice Test

Here's what you get

146	145
PRE-ASSESSMENTS QUESTIONS	POST-ASSESSMENTS QUESTIONS

Features

Each question comes with detailed remediation explaining not only why an answer option is correct but also why it is incorrect.

Unlimited Practice

Each test can be taken unlimited number of times until the learner feels they are prepared. Learner can review the test and read detailed remediation. Detailed test history is also available.

Each test set comes with learn, test and review modes. In learn mode, learners will attempt a question and will get immediate feedback and complete remediation as they move on to the next question. In test mode, learners can take a timed test simulating the actual exam conditions. In review mode, learners can read through one item at a time without attempting it.

13. Performance Based Labs

uCertify's performance-based labs are simulators that provides virtual environment. Labs deliver hands on experience with minimal risk and thus replace expensive physical labs. uCertify Labs are cloud-based, device-enabled and can be easily integrated with an LMS. Features of uCertify labs:

- Provide hands-on experience in a safe, online environment
- Labs simulate real world, hardware, software & CLI environment
- Flexible and inexpensive alternative to physical Labs
- Comes with well-organized component library for every task
- Highly interactive - learn by doing
- Explanations and remediation available
- Videos on how to perform

Lab Tasks

- Counting the Occurrence of Each Item in a Bag
- Performing Matrix Multiplication
- Transposing a Matrix
- Finding the Intersection of Two Arrays
- Handling an Exception
- Counting the Entries in a Bag
- Adding a Node at the End of a Doubly Linked Chain
- Removing First Node from a Doubly Linked Chain
- Adding Nodes to the Beginning of a Doubly Linked Chain
- Searching an Entry in a Bag
- Rearranging the Integers in an Array
- Creating a Stack and Adding Five Elements to it
- Removing an Element from a Stack
- Searching an Element in a Stack
- Transferring the Elements of One Stack to Another
- Transforming an Infix Expression to a Postfix Expression

- Evaluating a Postfix Expression
- Evaluating an Infix Expression
- Testing an Input String for Palindrome Using a Stack
- Creating an Array-Based Stack to Retrieve the Topmost Entry
- Creating a Vector-Based Stack to Retrieve the Topmost Entry
- Creating Custom Exception Class
- Creating a Queue
- Creating a Deque
- Creating a Priority Queue
- Removing the First Element from a Circular Linked Queue
- Removing the First Element from a Doubly Linked Deque
- Creating a Recursive Void Method to Print First Five Natural Numbers
- Traversing the Elements of a Linked List in Reverse Order
- Creating a Recursive Method to Return the Factorial of a Number
- Replacing an Element in the Linked List
- Removing an Element from the Specified Position in a Linked List
- Locating an Element in the Linked List
- Adding an Element at the Specified Position in a Linked List
- Converting an ArrayList to an Array
- Working with a List
- Traversing a List
- Removing Duplicates from the List
- Evaluating a Prefix Expression
- Sorting an Array using Selection Sort
- Sorting an Array using Insertion Sort
- Sorting an Array using Shell Sort
- Sorting an Array using Merge Sort
- Sorting an Array using Quick Sort
- Sorting an Array using Radix Sort
- Replacing a Word Using the StringBuilder Class
- Inserting an Element into a Sorted Array
- Using Polymorphism
- Using the Abstract Class
- Sorting a List using Inheritance
- Performing Sequential Search

- Performing Binary Search
- Implementing a Dictionary
- Implementing a Map
- Searching for a Key in a Dictionary
- Using Linear Probing in a Hash Table
- Implementing HashMap
- Traversing a Binary Tree using Inorder Traversal
- Traversing a Binary Tree using Preorder Traversal
- Traversing a Binary Tree Using Postorder Traversal
- Counting the Nodes of a Binary Tree
- Computing the Height of a Binary Tree
- Searching a Node in the Binary Search Tree
- Removing a Node from a Binary Search Tree
- Adding Nodes to a Max Heap
- Removing the Maximum Value Node from a Max Heap
- Sorting an Array using Heap Sort
- Adding Nodes to an AVL Tree
- Using the DFS Traversal
- Using the BFS Traversal
- Using Topological Sort
- Using Dijkstra's Algorithm
- Printing an Adjacency List
- Printing an Adjacency Matrix

Here's what you get

74

PERFORMANCE BASED LAB

14. Post-Assessment

After completion of the uCertify course Post-Assessments are given to students and often used in conjunction with a Pre-Assessment to measure their achievement and the effectiveness of the exam.

GET IN TOUCH:

3187 Independence Drive
Livermore, CA 94551,
United States

+1-415-763-6300

support@ucertify.com

www.ucertify.com